

计算机信息管理

毕业设计报告

**项 目 组： 第七组**

**专 业： 计算机信息管理**

**学 院： 应用技术学院**

**指导教师： 刘艳秋**

**实训时间：** 2021年3月1日—2021年6月4日

应用技术学院***CCUTSOFT***

**目 录**

[一、项目进度计划 1](#_Toc71844007)

[二、团队组成 1](#_Toc71844008)

[三、项目概况 1](#_Toc71844009)

[3.1 项目名称 1](#_Toc71844010)

[3.2 项目介绍 1](#_Toc71844011)

[3.3 代码结构 1](#_Toc71844012)

[3.4 项目流程 1](#_Toc71844013)

[四、程序设计与实现 1](#_Toc71844014)

[4.1 数据爬取 1](#_Toc71844015)

[4.2 微博热搜数据爬取 1](#_Toc71844016)

[4.3 世界各国疫情数据爬取 1](#_Toc71844017)

[4.4 Python与MySQL数据库交互 1](#_Toc71844018)

[4.5 Flask构建Python Web项目 1](#_Toc71844019)

[4.6 基于Echarts 图表展示 1](#_Toc71844020)

[4.7 前后端交互 1](#_Toc71844021)

[五、项目完成结果展示 1](#_Toc71844022)

[六、参考文献 1](#_Toc71844023)

[七、实训总结 1](#_Toc71844024)

# 一、项目进度计划

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **题目名称** | **开始时间** | **完成时间** |
| **1** | 项目环境准备与搭建 | 3.1 | 3.8 |
| **2** | 数据爬取 | 3.8 | 4.12 |
| **3** | 数据的清洗与持久化存储 | 4.12 | 5.14 |
| **4** | 开发前端网页并与后台数据联通 | 5.14 | 5.26 |
| **5** | 利用数据可视化技术呈现结果 | 5.26 | 6.1 |

# 二、团队组成

|  |  |  |  |
| --- | --- | --- | --- |
| **职能** | **学号** | **姓名** | **分工** |
| **组长** | 20184670 | 焦天 | 数据爬取分析&存储 |
| **组员** | 20184686 | 吕钊丞 | 前端网页优化 |
| 20184689 | 彭家豪 | 数据可视化 |

# 三、项目概况

3.1 项目名称

新型冠状病毒疫情实时监控系统

项目地址：<http://127.0.0.1:5000>

3.2 项目介绍

基于全球新型冠状病毒所引起的疫情风波以及目前全球疫情逐渐加重的大环境下，针对中国境内以及全球范围的疫情实时监控所开发的网页系统。该系统是基于Python + Flask + Echarts 所创造的。它能够实时监控并统计中国境内以及全球范围内的新冠病毒感染确诊人数、疑似感染人数、治愈人数、死亡人数等，同时以图表和地图的形式展现出来。

项目功能如下：

·统计全国各省市地区每日疫情数据，并以图表形式展现。

·统计全国疫情历史数据，并以图表形式展现。

·统计全球疫情历史数据，并以地图形式展现。

·统计微博热搜数据，并以词云图形式展现。

·统计全球各国疫情数据。

项目主要技术：

·Python网络爬虫

·Python与MySQL数据库交互

·Flask构建的Python Web项目

·Echarts数据可视化

3.3 代码结构

-毕业设计\

|-app.py —— Flask主程序

|-chromedriver.exe —— Chrome浏览器爬虫插件

|-nameMap.py —— 全球国家名称翻译

|-spider.py —— 爬虫文件

|-static\

| |-css\

| | |-main.css

| | |-style.css

| |-js\

| | |-china.js

| | |-controller.js —— 主控制类

| | |-echarts-wordcloud.min.js

| | |-echarts.min.js

| | |-ec\_center.js

| | |-ec\_left1.js

| | |-ec\_left2.js

| | |-ec\_right1.js

| | |-ec\_right2.js

| | |-jquery.js

| | |-vintage.js

| | |-world-controller.js

| | |-world-map.js

| | |-world.js

|-templates\

| |-main.html

| |-world.html

|-utils.py —— PyMySQL交互

3.4 项目流程

通过对腾讯疫情数据的爬取，并持久化将其保存在数据库当中。最后通过前端网页展示，将所获取的数据进行可视化展现。

# 四、程序设计与实现

4.1 数据爬取

4.1.1 爬虫概述

Python的运用当中，爬虫是不可缺少的一部分，也是其语言出色的一大作用。爬虫是一个向网站发起请求，并从网站响应中提取所需要的数据的自动化程序。

爬虫基本流程如下：

1. **向网站发起请求，获取网站的响应**

通过一些常用库（urllib、urllib3、requests）等对目标站点进行请求，等同于自己打开浏览器，输入网址访问。服务器返回的响应内容的格式通常为：html、二进制文件（视频、音频）、文档、Json字符串等。

1. **解析网站响应的内容(数据清洗)**

寻找自己需要的内容，就是利用正则表达式或者其他库提取目标信息

常用库（re、beautifulsoup4）

1. **保存数据至数据库中**

将解析得到的数据持久化到文件或者数据库中

4.1.2 爬取腾讯疫情数据

利用谷歌浏览器开发者选项(F12)分析腾讯疫情数据网站，经过解析，发现网站上所有疫情的数据都来源于以下几个URL对应的API接口，由于网站提供了数据的 API 接口，可以测试得到数据为 json 格式，所以我们只需要调用 json 库中相应的方法就可以完成数据解析。

1. https://view.inews.qq.com/g2/getOnsInfo?name="disease\_h5"   *# 全国各省市地区每日疫情数据*
2. https://view.inews.qq.com/g2/getOnsInfo?name="disease\_other"  *# 全国疫情历史数据*
3. https://api.inews.qq.com/newsqa/v1/automation/foreign/daily/list?country=国家名字 *# 国外各国疫情数据*

4.1.3 数据分析

从4.1.2中分析可得，虽然成功获取到了数据，但是内容无序不可以直接使用，所以需要从响应后进行解析，从中提取所需要的数据集，并且根据这些数据集进行数据库的构建。

1. *# 设置请求头，防止爬虫失败*
2. headers = {
3. "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36"
4. }
5. *# json转换格式*
6. r1 = requests.get(url1, headers)
7. r2 = requests.get(url2, headers)
8. *# 将json格式转换为字典格式*
9. res1 = json.loads(r1.text)
10. res2 = json.loads(r2.text)
11. data\_all1 = json.loads(res1["data"])
12. data\_all2 = json.loads(res2["data"])

我们可以发现API返回的数据格式为Json格式，所以我们可以利用python第三方库将Json格式转化为python中我们熟悉的字典格式，然后利用熟悉的字典键值对相关操作获取我们想要的数据。

4.2 微博热搜数据爬取

对微博热搜数据的爬取并分析后，发现出现错误并且数据为乱码，检查原因后发现，微博热搜数据网页页面使用了动态渲染技术，普通的爬虫无法正常获取数据。

4.2.1 selenium介绍

selenium是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。要想利用该工具，还需要你安装对应的浏览器（谷歌、火狐）和浏览器驱动

selenium 使用步骤：

* 创建浏览器对象
* 浏览器.get()
* 浏览器.find()

4.3 世界各国疫情数据爬取

爬取各国疫情数据具体操作步骤如下：

* + - * 将世界各国国家名字存在数组中
      * 遍历数组，将国家名字拼接至API接口中
      * 对爬取数据进行解析整理，持久化保存到数据库中

4.4 Python与MySQL数据库交互

4.4.1 数据库初始化

使用 pymysql 模块与数据库交互，步骤如下：

* 建立连接
* 创建游标
* 根据游标执行操作
* 关闭连接
* def get\_conn():
* *# 建立连接*
* conn = pymysql.connect(host="127.0.0.1", user="root", password="qweasd1234", db="cov", charset="utf8")
* *# 创建游标*
* cursor = conn.cursor()
* return conn, cursor

4.4.2 数据库建表

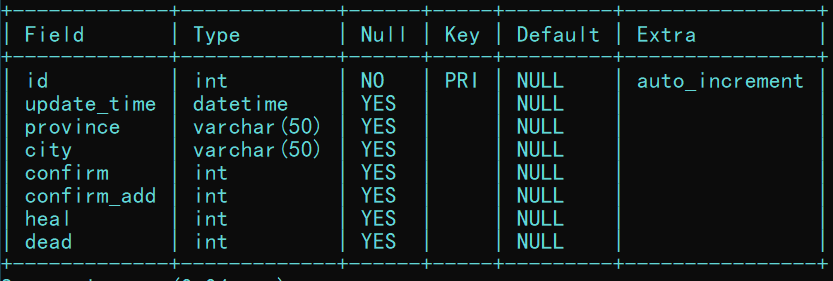
根据爬取的腾讯疫情数据、百度热搜数据以及世界各国疫情数据，我们需要在数据库中建立四张表格，分别名为details(各省市疫情数据)、fforeign(世界各国疫情数据)、history(中国疫情历史数据)、hotsearch(热搜数据)，数据库中四张表格如下：

4 - 数据表图

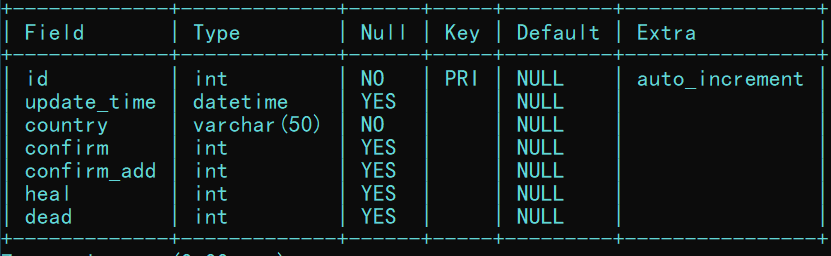


Details 表字段如下：

4 - details表

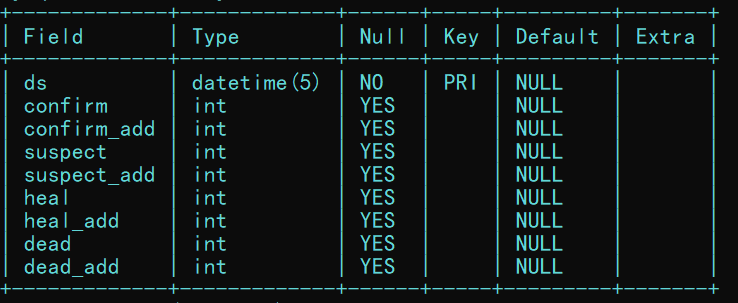


4 - fforeign 表



Fforeign 表字段如下：

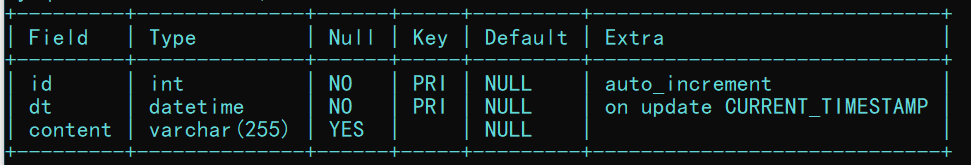
4 - history 表



History 表字段如下：

Hotsearch 表字段如下：

4 - hotsearch 表



4.4.3 数据库交互

数据持久化过程中最难是建表，在建表的过程中需要考虑主键以及其他属性等等，而与数据库交互的过程就是CURD逻辑代码的书写。唯一要注意的就是数据库中时间字段的类型与python中时间字段类型是不一致的，所以在**与数据库交互之前，必须进行类型的强制转换**

4.5 Flask构建Python Web项目

4.5.1 Flask介绍

Flask是一个基于Python语言且遵守WSGI协议的轻量级Web开发框架，非常适合小项目的编写，只需要具备基本的python开发技能，就可以开发出一个web应用来。

相比于Django, Flask算的上是微型的Web框架了，它只有路由和模板渲染两个功能，想干别的事都需要使用插件。

和Django大包大揽不同，Flask建立于一系列的开源软件包之上，这其中最主要的是WSGI应用开发库[Werkzeug]

主要特点小而轻，属于短小精悍型框架，同时入门简单

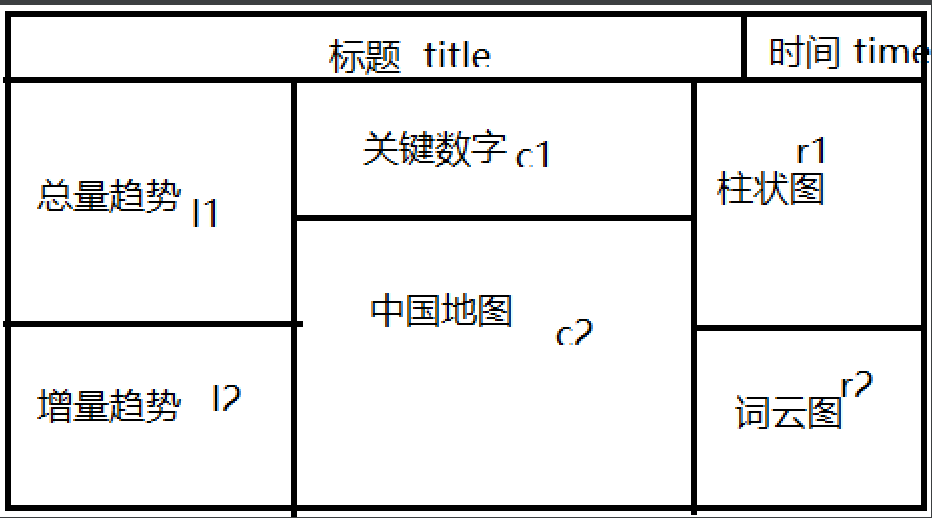
4.5.2 Flask运行

1. *#导入 Flask 类*
2. from flask import Flask
3. *# 接着我们创建这个类的实例。第一个参数是应用模块或者包的名称。如果你使用一个单一模块（就像本例），那么应当使用 \_\_name\_\_ ，因为名称会根据这个模块是按应用方式使用还是作为一个模块导入而发生变化（可能是'\_\_main\_\_' ，也可能是实际导入的名称）。这个参数是必需的，这样 Flask就可以知道在哪里找到模板和静态文件等东西。更多内容详见Flask文档。*
4. app = Flask(\_\_name\_\_)
5. *# 然后我们使用#route()装饰器来告诉 Flask 触发函数的 URL，这里绑定到了根目录，意思就是说当输入IP和PROT之后就可以访问hello\_world函数 。*
6. @app.route('/')
7. *# 函数名称可用于生成相关联的 URL ，并返回需要在用户浏览器中显示的信息。*
8. def hello\_world():
9. return 'Hello World!'
10. *# 最后，使用 run() 函数来运行本地服务器和我们的应用。if\_\_name\_\_== '\_\_main\_\_': 确保服务器只会在使用 Python 解释器运行代码的情况下运行，而不会在作为模块导入时运行。*
11. if \_\_name\_\_ == '\_\_main\_\_':
12. app.run()

4.5.3 前端布局

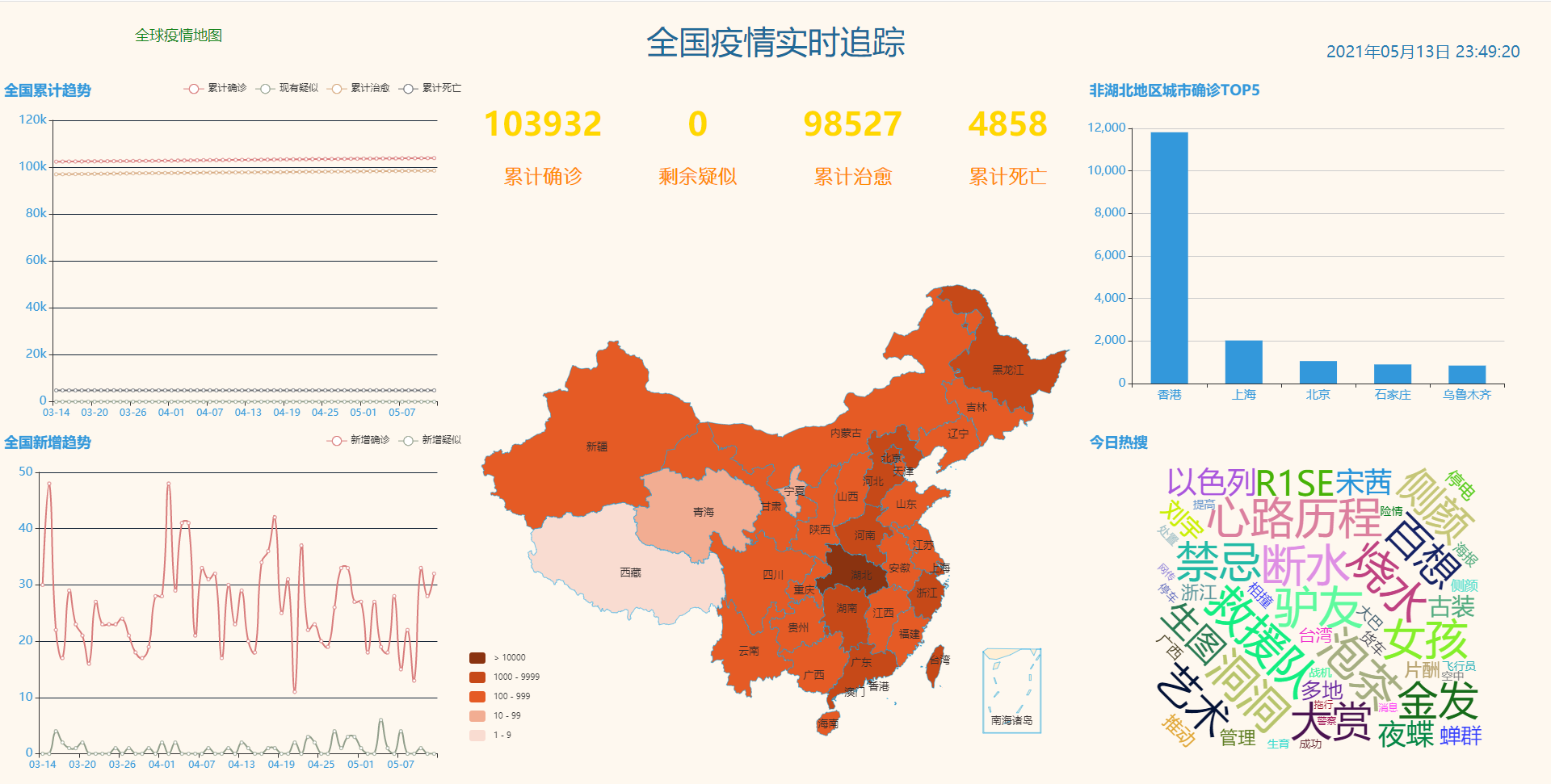
前端页面布局主要利用CSS语言进行编写，根据下面绘制的简单页面规划图以及基本的CSS语句即可完成前端页面布局的书写。

4 - 6 页面布局



基于以上原理，最终完成的效果图如下：

4 - 7 最终完成图



对于4-7完全图，主要包含有：

* 中国地图
* 左上累计趋势图
* 左下新增趋势图
* 右上最高确诊城市图
* 右下热搜词云图
* 全球疫情地图链接
* 标题
* 数据统计图
* 当前时间

4.6 基于Echarts 图表展示

4.6.1 Echarts 原理

ECharts，缩写来自Enterprise Charts，商业级数据图表，是百度的一个开源的数据可视化工具，提供了丰富的图表库，能够在PC端和移动设备上流畅运行。在前端代码中引入Echarts图表是非常简单的，引入Echarts图表的代码如下：

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. *<!-- 引入 ECharts 文件 -->*
6. <script src="echarts.min.js"></script>
7. </head>
8. </html>

4.6.2 插入中国地图

分析可得，目前我们需要的图表一共有六个：\*\*词云图、中国地图、世界地图、3个柱状图或者折线图。这些图表的js文件在网上都有相应的资源，下载后引入前端界面即可。

以中国地图为例：

导入china.js

复制中国地图option，加以整理至ec\_center.js文件中。

剩余几个图表的操作过程和原理与引入中国地图相同。

4.7 前后端交互

4 - 8 效果完全图

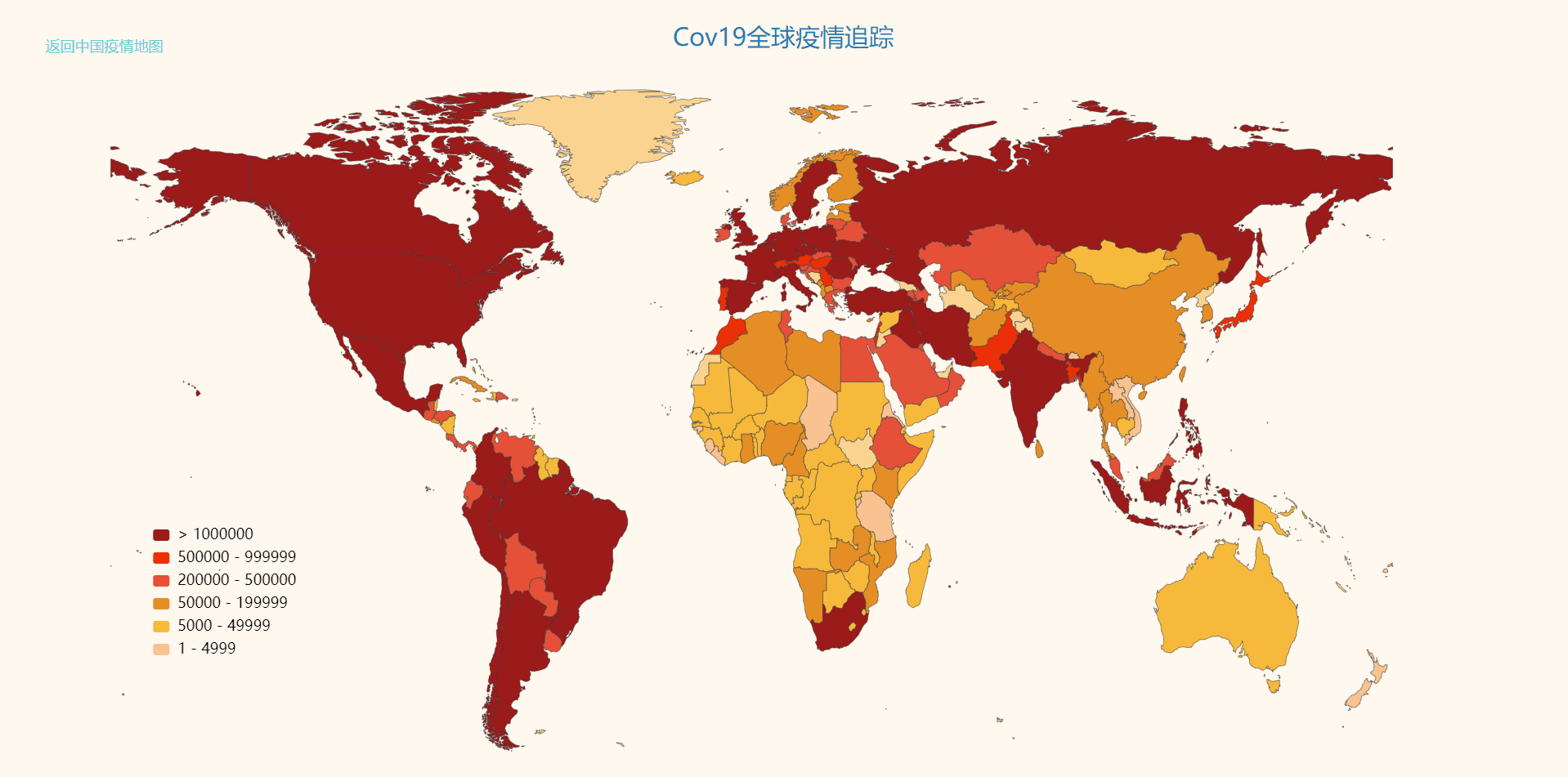
4.7.1 Ajax介绍

Ajax 是 Asynchronous JavaScript and XML 的简称，通过 Ajax 向服务器发送请求，接收服务器返回的 json 数据，然后使用 JavaScript修改网页的来实现页面局部数据更新。Ajax的基本格式如下：

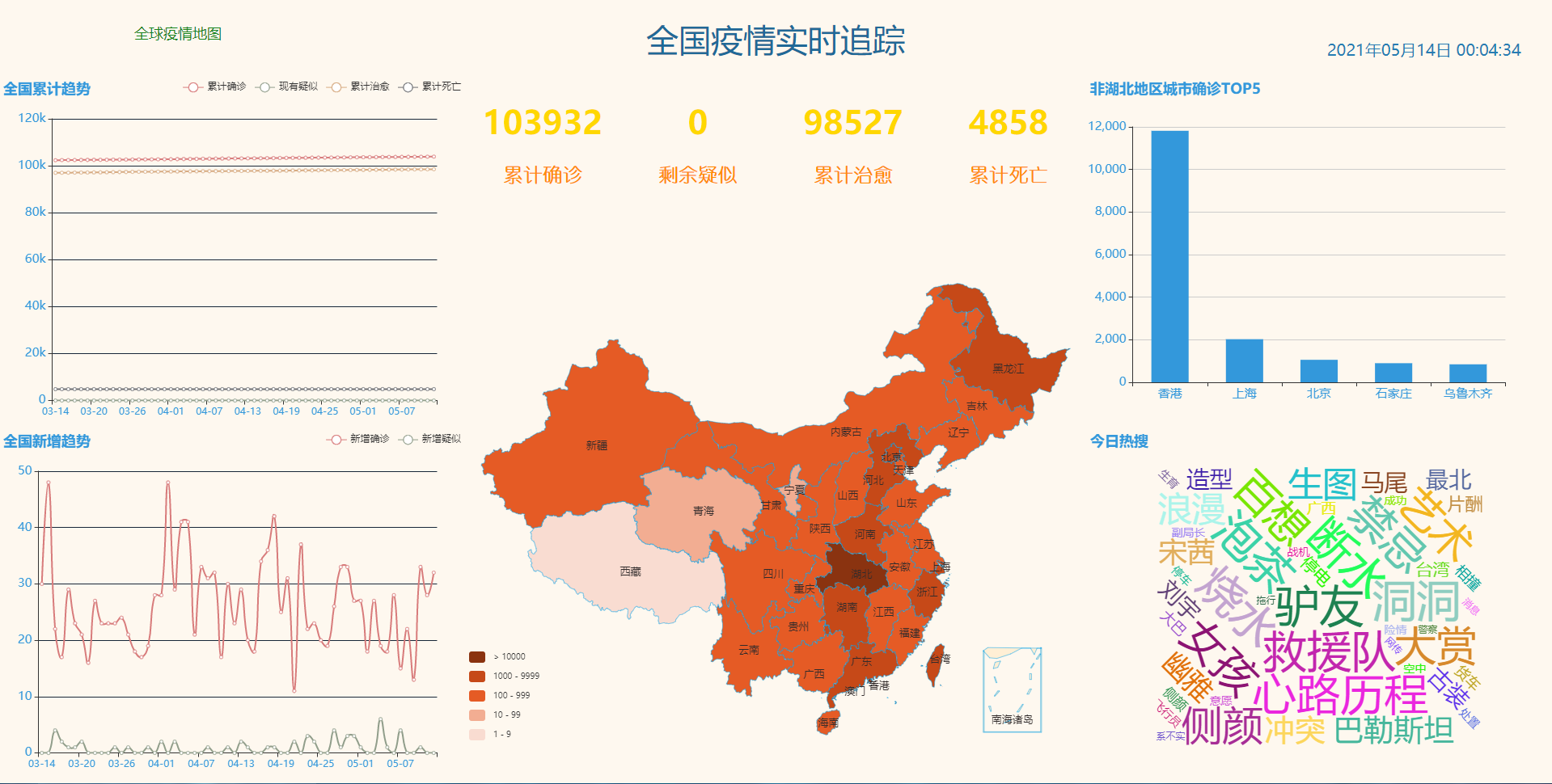
1. $.ajax({
2. type:"post",  *// 请求类型*
3. url: "/",  *// 请求地址*
4. timeout: 10000,   *// 超时时间*
5. data:{"k1":"v1","k2":"v2"},  *// 数据*
6. success: function(datas) {
7. *// 请求成功的回调函数，datas为返回数据*
8. },
9. error: function() {
10. *// 失败时请求执行*
11. }
12. })

根据Ajax原理以及前文实现的数据库相关操作，即可以完成前端页面数据的填写。

# 五、项目完成结果展示



5 - 1 项目完成主图



5 - 2 世界完成图

# 六、参考文献

Echarts 官方教程：[Documentation - Apache ECharts](https://echarts.apache.org/zh/tutorial.html#5%20%E5%88%86%E9%92%9F%E4%B8%8A%E6%89%8B%20ECharts)

WSGI 协议介绍：[尝试理解Flask源码 之 搞懂WSGI协议 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/46983059)

Flask 官方文档：[快速上手 — Flask 中文文档（ 1.1.2 ） (dormousehole.readthedocs.io)](https://dormousehole.readthedocs.io/en/latest/quickstart.html#quickstart)

Selenium & Python应用：[selenium + python自动化测试环境搭建 | selenium中文网](http://www.selenium.org.cn/1203.html)

# 七、实训总结

|  |
| --- |
| 小组总结 |
| 在项目的完成过程中，学习了一些常见的爬虫、反爬虫、反反爬虫等常见技术，不仅开拓了视野，同时对于爬虫整个流程和原理都有了更加深刻的认识，相信以后写爬虫程序应该能够更加的得心应手。  虽然说整个项目的前面页面非常简单，但是在学习的过程中，还是掌握了例如HTML,CSS,JS等基础知识。  除了前端页面的处理，更多的是对各类主流网站进行爬虫操作以及存储到数据库操作，这不仅加深了我们对数据库基础知识的认知，更加强了书写数据库操作语言的熟练度。  为了使得前端页面更加美观，我们尝试了三种不同的主题，最后敲定了亮色系以便观察。同时也对各种第三方库进行了多种比较，确认了最便捷、最高效率的一些常用库。  虽然这个项目仅仅处于练手阶段，但对于现阶段学习的我们来说，这是另一次更加值得骄傲的项目的学习。不仅对以前的知识进行巩固，更对一些从没接触过的技术而了解，尽管还有很多不足的地方，但我们相信，秉持着这种热情和严谨，我们今后的项目会越来越优异，同时我们也会变得越来越优秀。  签字：Matrix Revolution  2021年 6月 1日 |

|  |
| --- |
| 个人总结 |
| 这是我在长春工业大学专科三年的最后一次项目，尽管有些不舍，但还是将自身的热情和认真将这次项目做到令我自己满意的程度。  项目采用主流的Mysql数据库作为数据的持久化选择，项目中设计到大量的CURD操作，需要书写大量的SQL语句，这不仅加强了我对Mysql理论知识的熟悉，更是在实践中训练了书写SQL语句的能力。  为了使爬虫达到最快的爬取速度，我查阅了网上相关资料和一些第三方库来进行比较，有些常规爬虫对常规网页的爬取没有任何问题，但是对于热搜数据等爬取后的结果分析显示，大多数数据都是json格式而无法直接用字典等进行分割并存储。在接触了selenium后发现，还有许多爬虫更加高级且更加多用途化。  整个项目的研究程度大于书写代码，换句话说，在这个项目中更偏向于学习各类知识，而不是单纯的为了完成项目而去学习。在认识到了项目所需要的必须知识后，我更偏向于深一步挖掘如何进行、如何操作、如何实现，而不是将就完成而已。这相比于之前的项目学习，这是我最大的一个突破。  尽管这是这个团队最后一次的项目设计，但我会将这次项目保留，以便在以后的学习或项目创作中遇到困难的时候，回想起当初的初心和当时的状态。  也是对我三年学习生涯最后的一个交代。  祝愿我的朋友们各自安好。  签字：焦天  2021年 6月 1日 |
| 个人总结 |
| 在学习Python的过程中我了解到了Python的数据处理能力和爬虫，数据处理不仅仅需要Python语言的支持，更多的是分析数据的能力，爬虫呢相对来说就比较简单了，因为之前也接触过一点Web的知识，所以对于爬虫的理解也不是太过难，但是还是有一些概念绕的我头有点晕。而且现在网上的爬虫教程在版本信息或者demo的讲解够全面，所以看得有些困难。  在此次实训期间，我和我的学习小组组员们也尝试着开发了一个爬虫程序。我们开发的程序用来爬取百度文库中的文章，因为目前我们学校暂停了与百度文库的合作关系  在这短短的时间里，让我深深的感觉到自己在实际应用中所学专业知识的匮乏。让我真真领悟到“学无止境”这句话的涵义。而老师所讲的，都是课本上没有而对我们又非常实用的东西，这又给我们的实训增加了浓墨淡采的光辉。我懂得了实际生活中，专业知识是怎样应用与实践的。在这些过程中，我不仅知道了职业生涯所需具备的专业知识，而且让我深深体会到一个团队中各成员合作的重要性，要善于团队合作，善于利用别人的智慧，这才是大智慧。靠单一的力量是很难完成一个大项目的，在进行团队合作的时候，还要耐心听取每个成员的意见，合作带来的往往是更完美的成果。  除此课本上的知识毕竟有限。通过实训，大家都有这样一个感觉，课本上的理论知识与实际工作有很大差距，只有知识是远远不够的，专业技能急需提高。  签字：吕钊丞  2021年 6月 1日 |
| 个人总结 |
| 通过这半个月的实训，我们都收获颇丰，总体来说对这次实训还是很满意的。尽管实训很累，但真的很感谢学校能够提供我们这样好的实训机会。  我们深刻的了解到，只有经历过，才知道其中的滋味。对于我而言，喜欢体验生活，通过这次实训，不得不说，Python真的是一门非常不错的语言，但是如果想要深入学习编程知识的话，不建议以Python作为入门语言，是能在Java或C的基础上去学习，这样会好很多。  而且现在随着人工智能的火热，Python必将会在未来的生产工作中占据很大比重。  所以，现在学习一些Python的知识还是很有必要的。而对于走向工作岗位的我们更要不断加强自己的专业技能，社会不会要一个一无是处的人，所以我们要更多更快的从一个学校人向社会人转变。  为此我们将会在以后的日子里继续努力，不断激励经验，不断磨砺自己，早日走向工作岗位。  签字：彭家豪  2021年 6月 1日 |